University of Málaga

Health Engineering

Laboratory Task

Search

Author

Alejandro Domínguez Recio

Course

Intelligent Systems

Teachers

Enrique Domínguez Merino

Jesús de Benito Picazo

Introduction

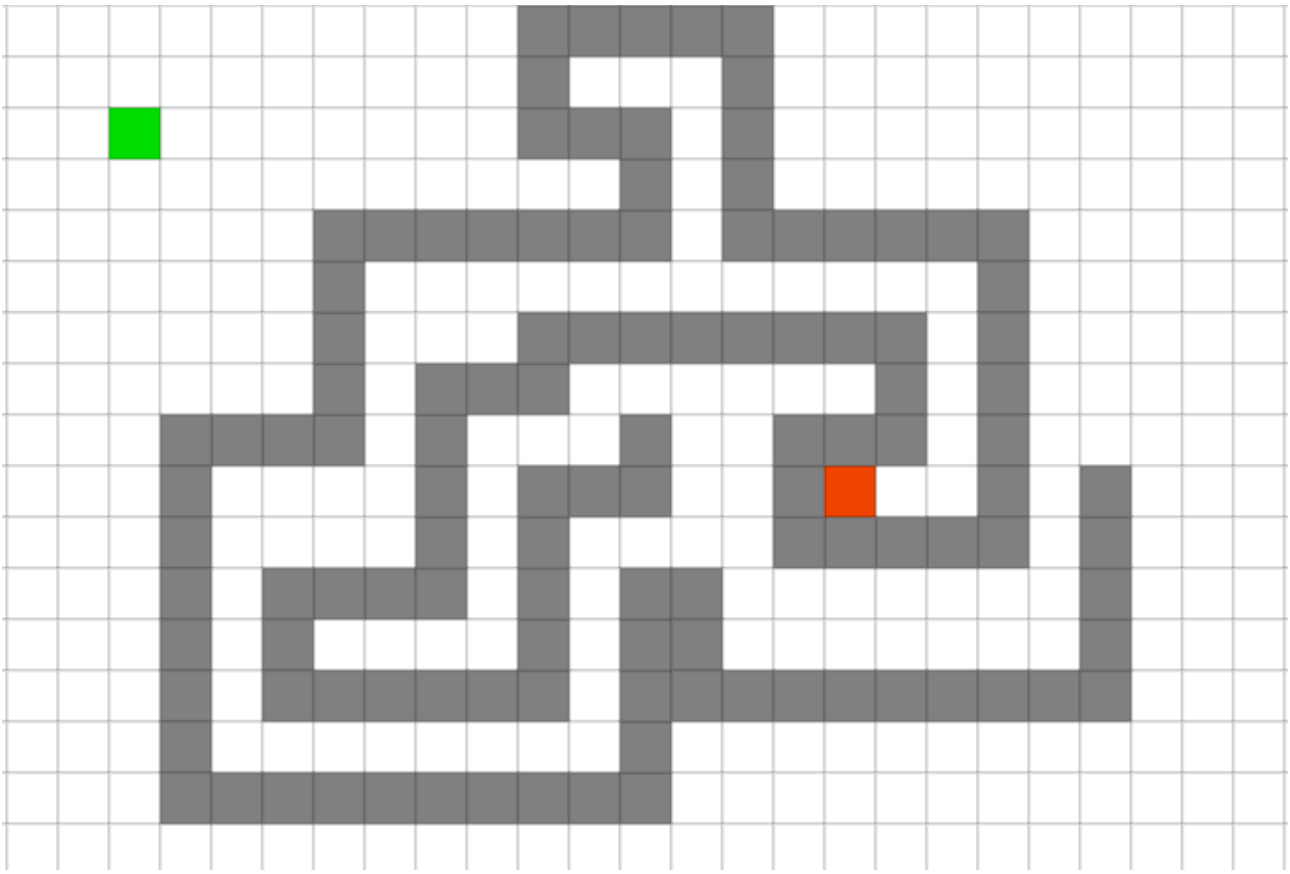
In this practice we are going to study the behavior of various search algorithms, for which we are going to use the PathFinding application.

We will experience the A * with Manhattan, Euclidean, octile and Chebyshev heuristics algorithms; breadth-first search; best-first search with Manhattan, Euclidean, octile and Chebyshev heuristics; and Dijkstra. For each algorithm we will detail the length of the obtained path, the elapsed time and the number of executed operations.

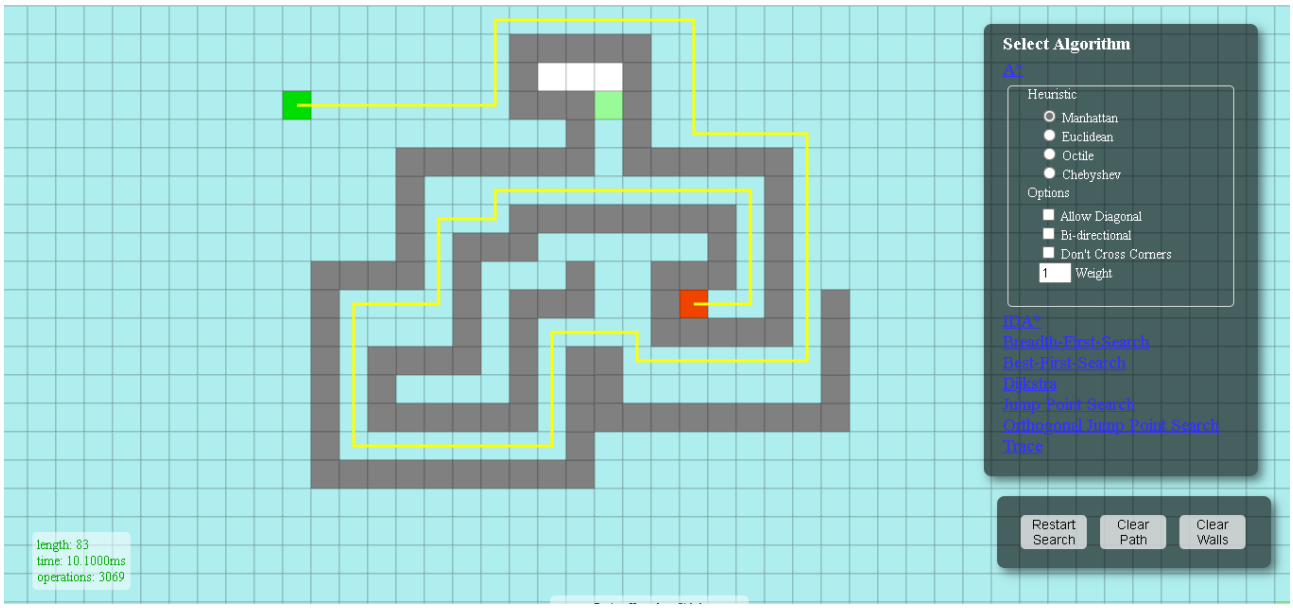
Procedure

To do the different studies we will have the following structure.

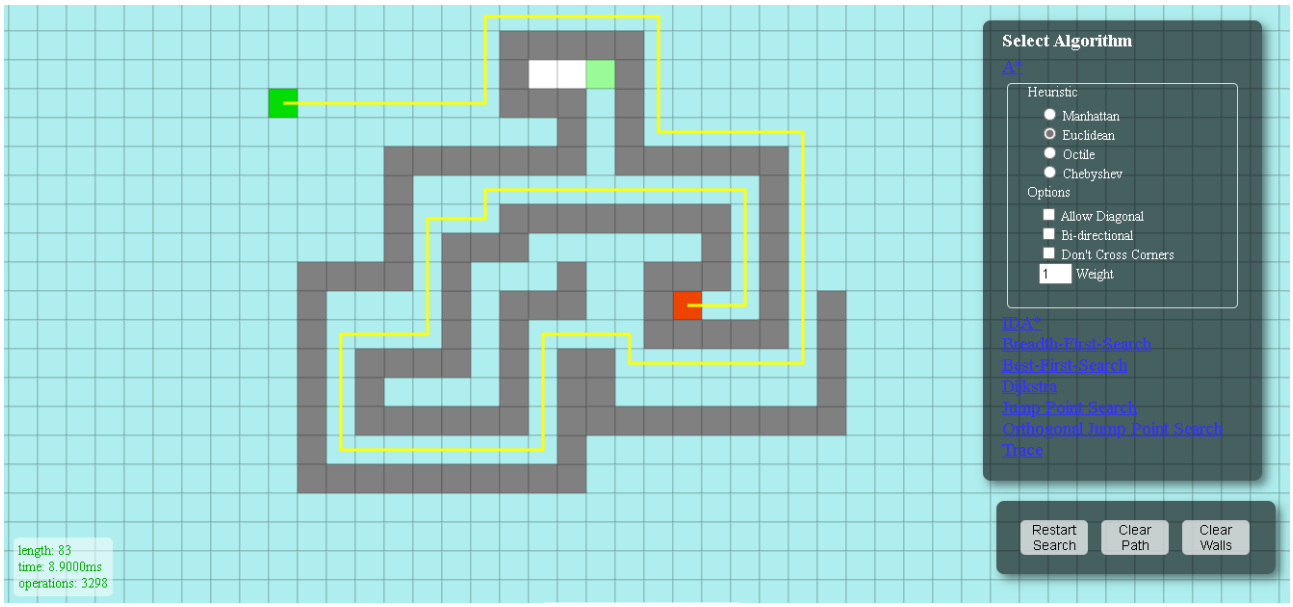
➔ **Maze**



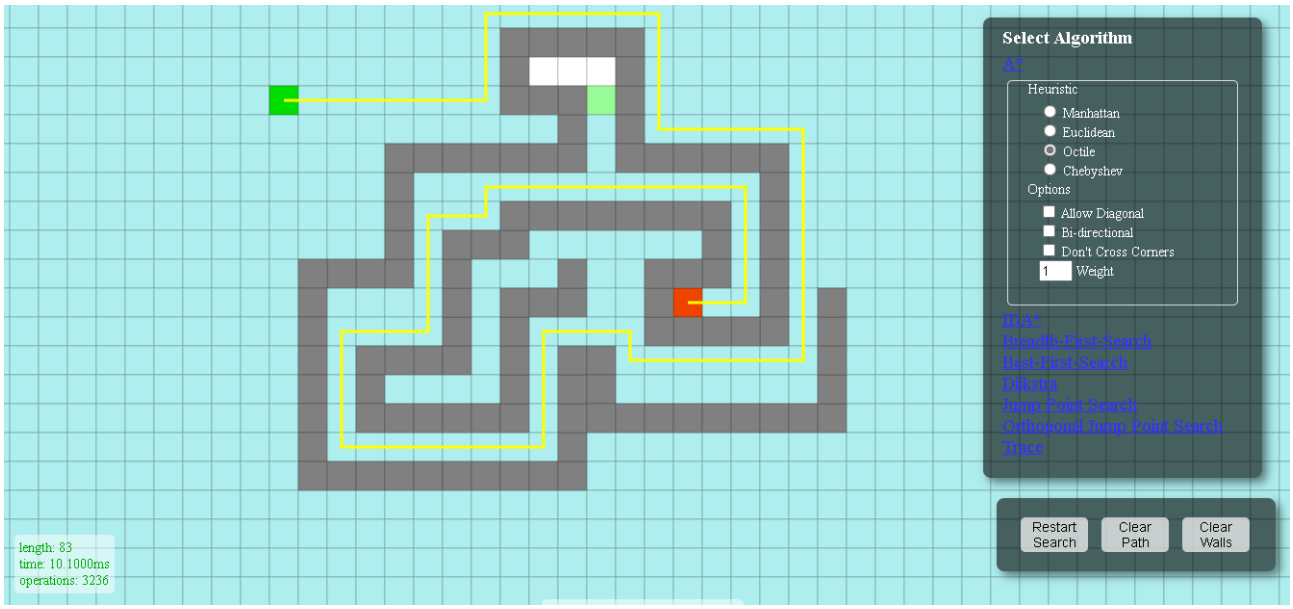
➔ A* with Manhattan



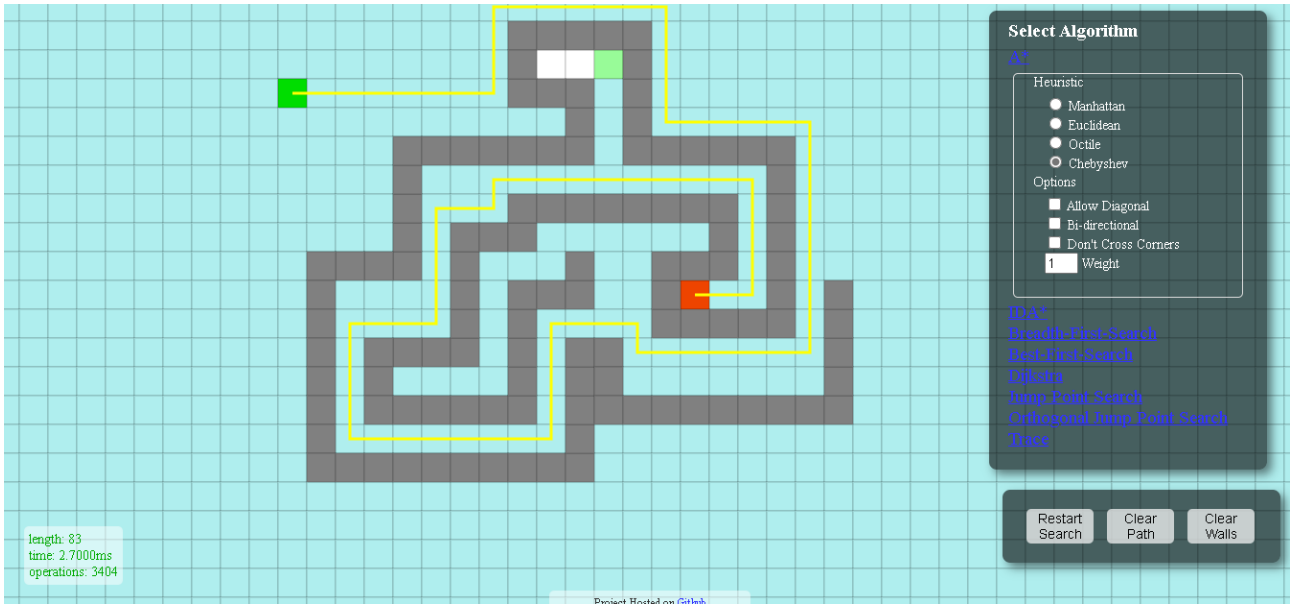
➔ A* with Euclidean



➔ A* with Octile

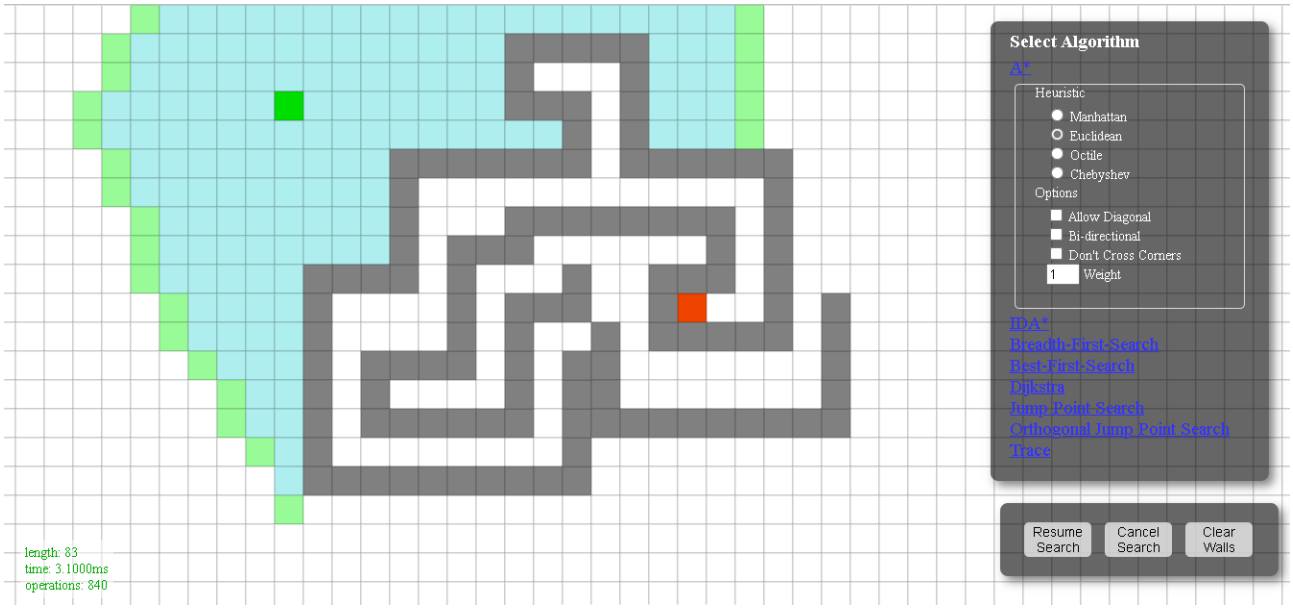


➔ A* with Chebysheb

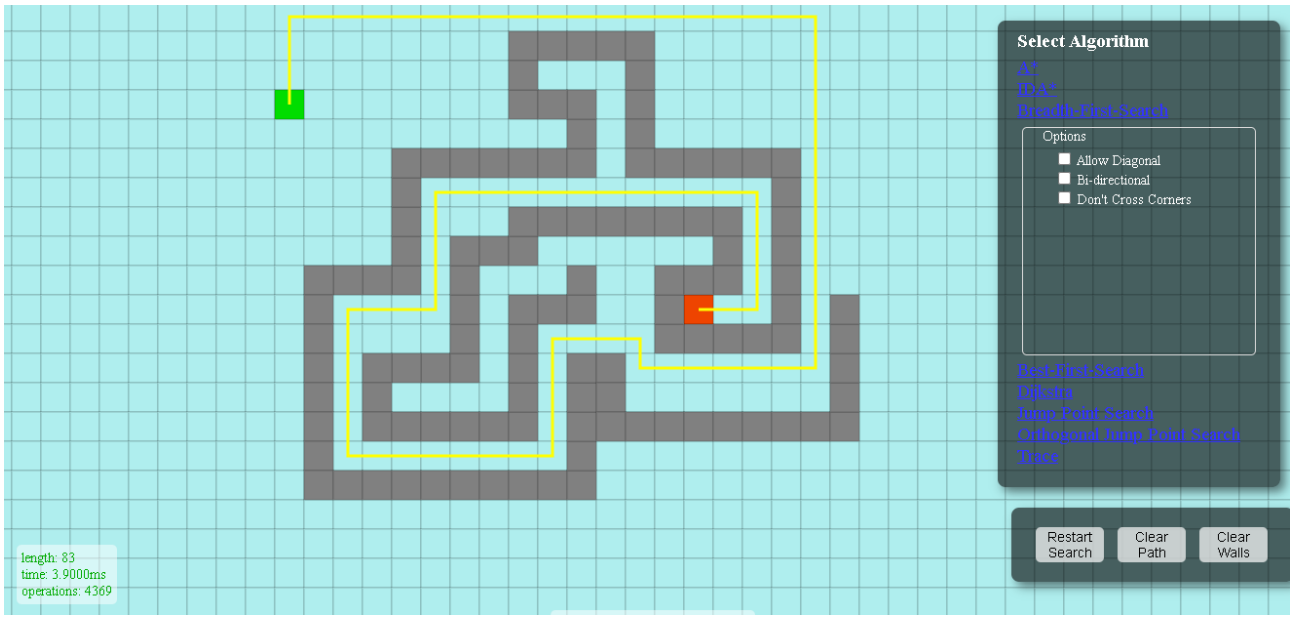


➔ Observations A*

As we can, the search space of A * has an expansion by areas that do not lead in an optimal way to the solution. This strategy causes a greater number of operations and taking into account that these operations, taking into account the heuristics, are more expensive, this is reflected in the total execution time.

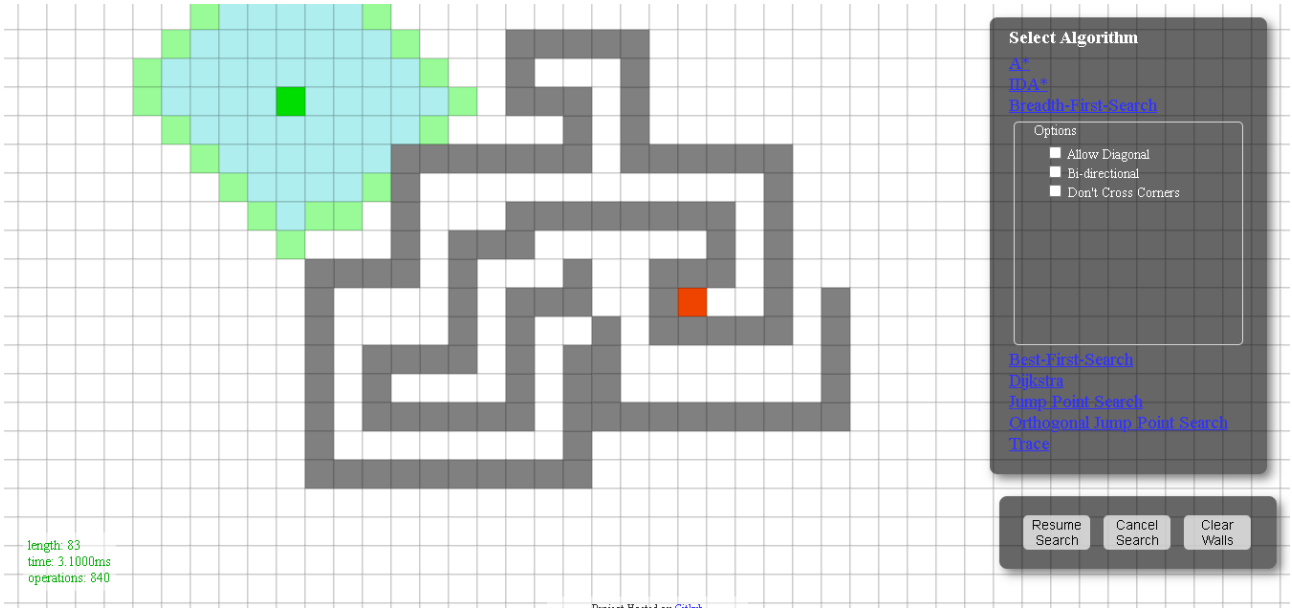


➔ Breadth-first search

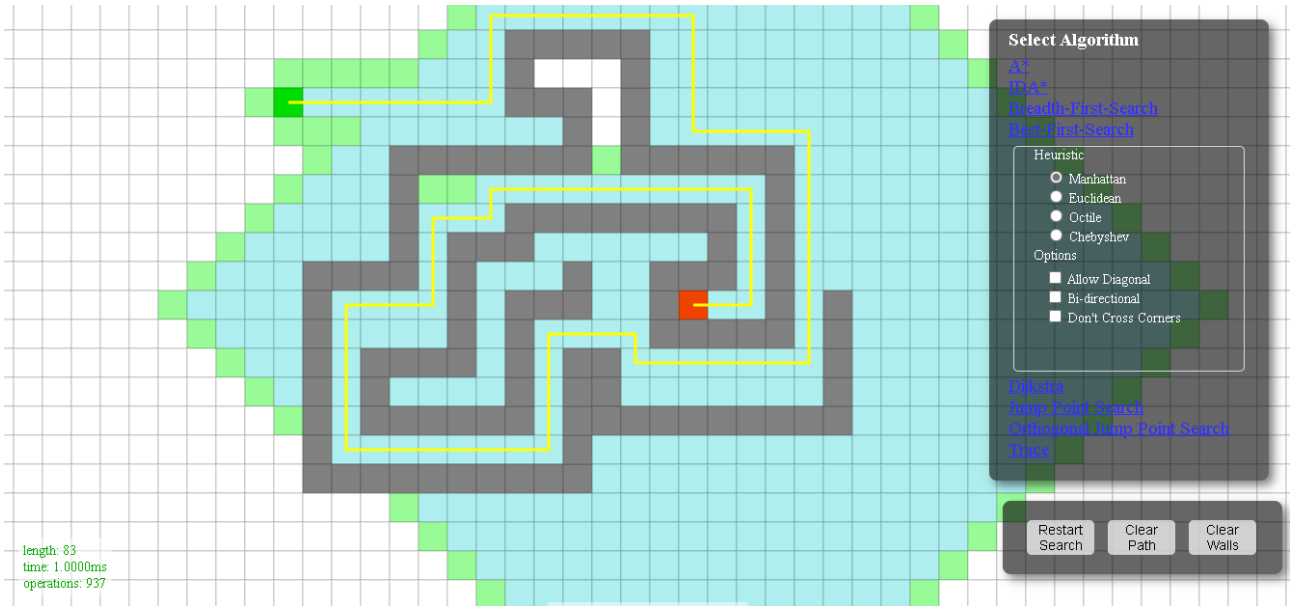


➔ Observations Breadth-first

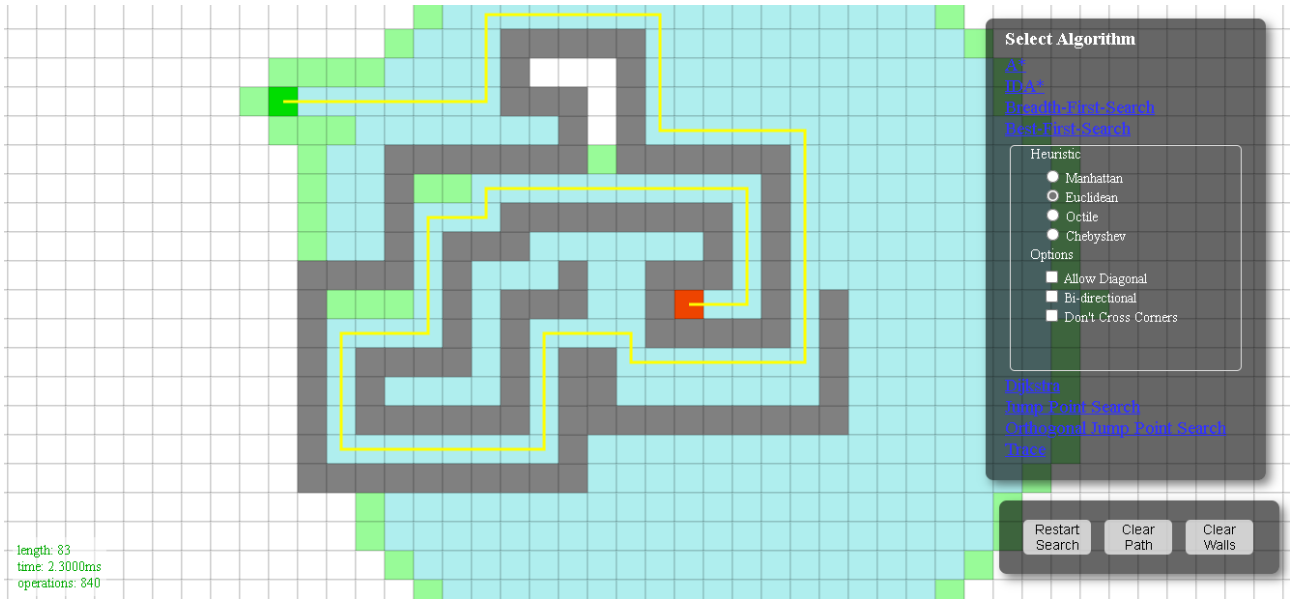
As we can, the search space of breadth first has an expansion by zones that do not lead in an optimal way to the solution in a similar way to what happened with A *. In this case, it was to be expected due to the characteristics of said algorithm. However, the number of operations is less due to the absence of heuristics.



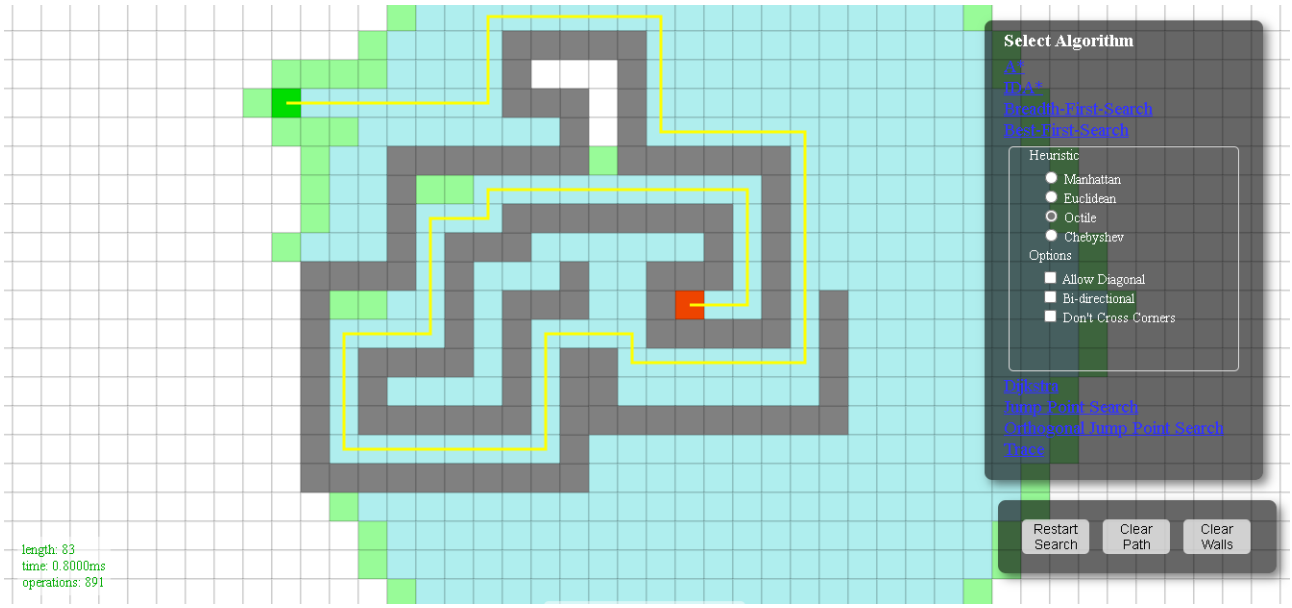
➔ Best-first with Manhattan



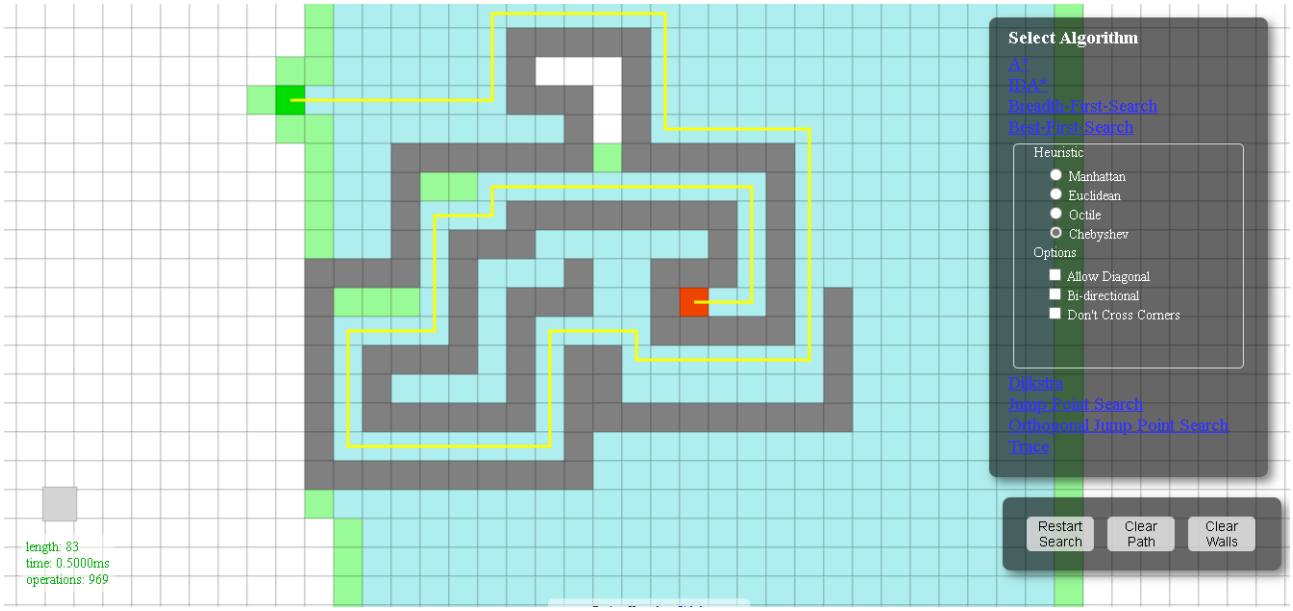
➔ Best-first with Euclidean



➔ Best-first with Octile

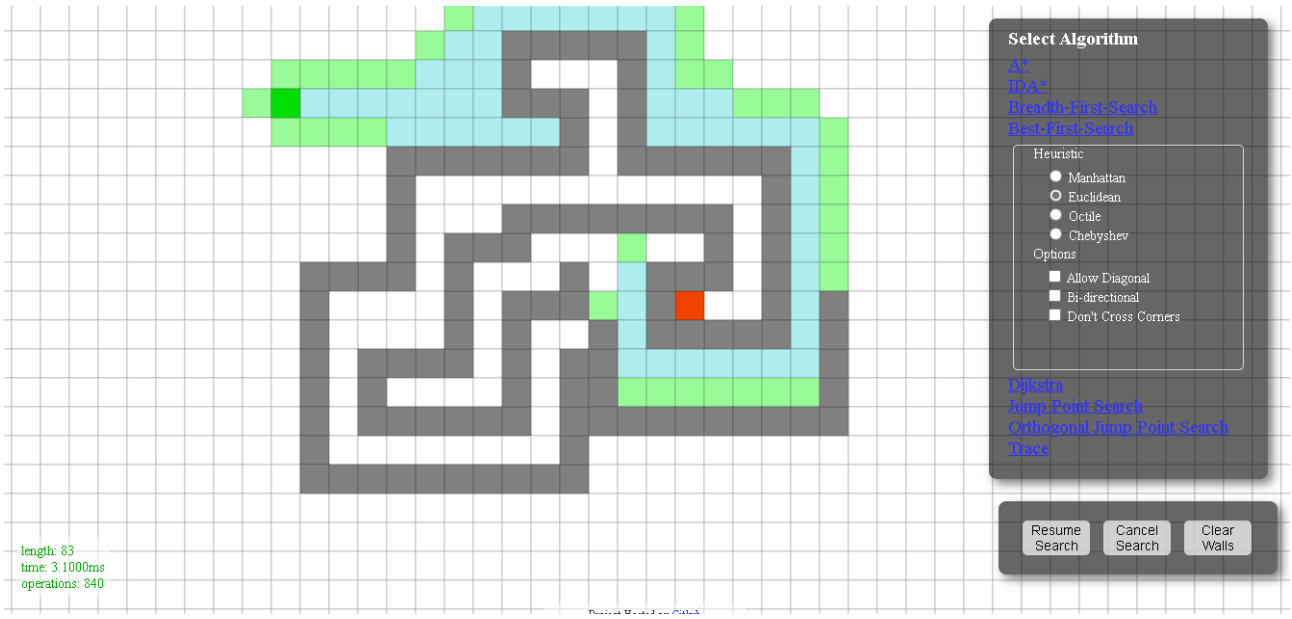


➔ Best-first with Chebyshev

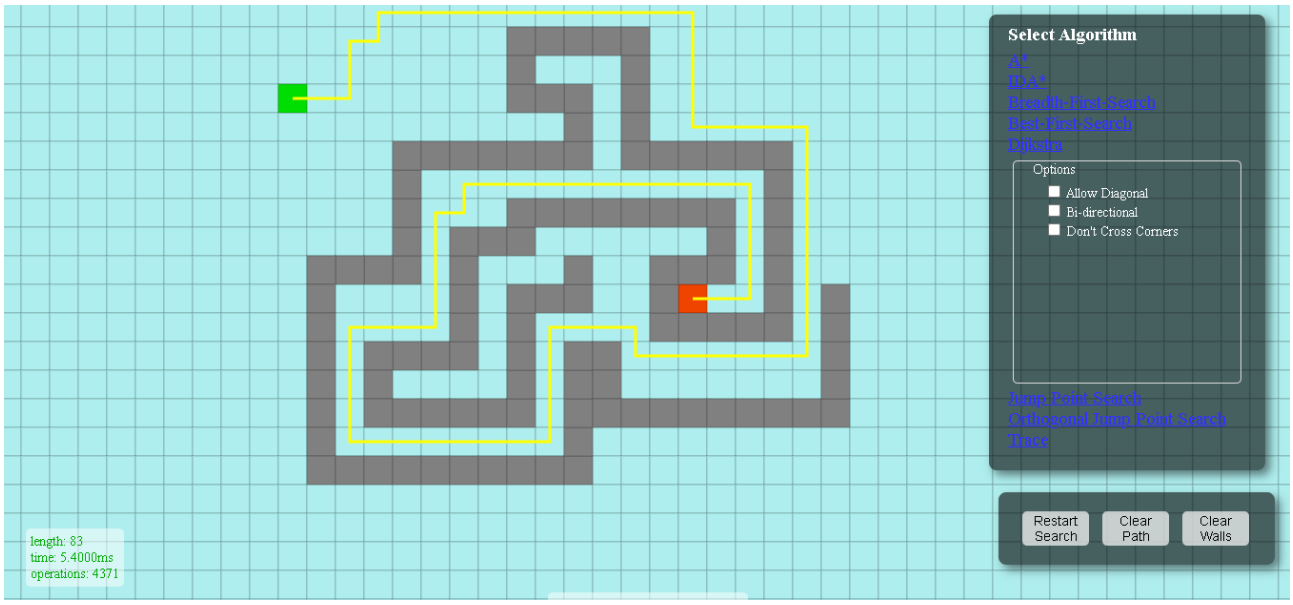


➔ Observations Best-first

As we can, the search space of best-first has a much more precise expansion, avoiding explorations of not optimal nodes. Clearly the success of this strategy is represented both in execution time and in operations carried out.

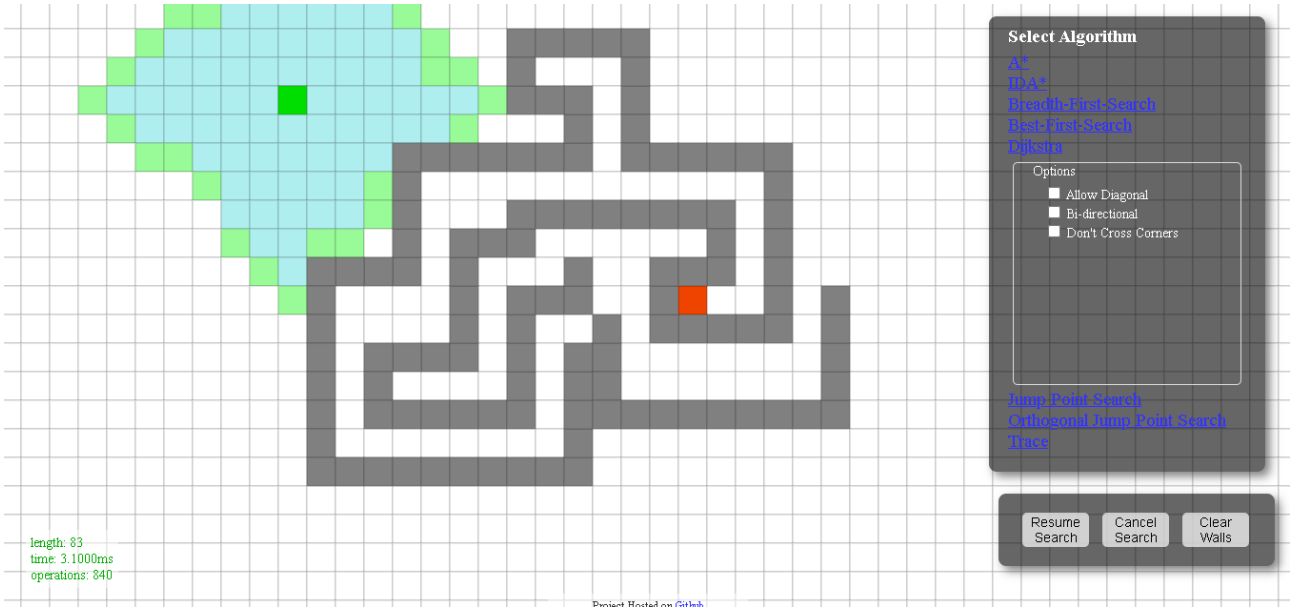


➔ Dijkstra



➔ Observations Dijkstra

As we can see with dijkstra, something similar to breadth first happens, the expansion to all reachable nodes causes an increase in the search space and therefore the number of operations and execution time.



➔ Results

Algorithm	Length	TimeMs	Operations
A*Manhattan	83	10,100	3069
A*Euclidean	83	8,900	3298
A*Octile	83	10,100	3236
A*Chebyshev	83	2,700	3404
Breadth-first	83	3,900	4369
Best-firstManhattan	83	1,000	937
Best-firstEuclidean	83	2,300	840
Best-firstOctile	83	0,800	891
Best-firstChebyshev	83	0,500	969
Dijkstra	83	5,400	4371

➔ Conclusion

By way of conclusion we can say that algorithms without heuristics provide us with a better result both in time and in operations compared to most of the opposite performance measures. To all this, adding that Dijkstra is the one that has carried out the most operations due to its operating characteristic of exploring all reachable unvisited nodes, although in execution time it is faster than A *, perhaps due to the need for heuristic calculation. Speaking a little more about A * we can say both by the nature of the problem, by the starting position of this or by the heuristic value associated with each node, the number of operations performed are close to those given by the most, Dijkstra, a in turn influencing the execution time that if they have been the highest due to the trigonometric calculations of its distance function. In relation to the one that has given us the best Best-first result, we can say that the initial delimitation of its search space which provided precision on the way to the solution, denoting this in the number of operations carried out and therefore its effect over time. of execution.